CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

A Project Report on

Network Dashboard for SDWAN

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science and Engineering

by

Deepanshu Goyal 1660326

Under the Guidance of

Chinthakunta Manjunath

and

Gururajan Sundar Rao

Department of Computer Science and Engineering

School of Engineering and Technology, CHRIST (Deemed to be University),

Kumbalgodu, Bangalore - 560 074

April-2020

CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

School of Engineering and Technology

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that **Deepanshu Goyal** has successfully completed the project work entitled "**Network Dashboard for SDWAN**" in partial fulfillment for the award of **Bachelor of Technology** in **Computer Science and Engineering** during the year **2019-2020**.

Chinthakunta Manjunath	**Gururajan Sundar Rao**
Assistant Professor	Technical Leader
Dr. K Balachandran	**Dr. Iven Jose**
Head of Department	Dean

CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

School of Engineering and Technology

Department of Computer Science and Engineering

BONAFIDE CERTIFICATE

It is to certify that this project titled ”Network Dashboard for SDWAN” is the bonafide work of

Name	Register Number
Deepanshu Goyal	1660326

Examiners [Name and Signature]	Name of the Candidate :
1.	Register Number :
2.	Date of Examination :

Head of Department
Computer Science, School of Engineering,
CHRIST (Deemed to be University),
Kumbalgodu, Bangalore,
Karnataka, India – 560074

Thursday, April 16, 2020

This is to certify that Mr. Deepanshu Goyal is currently undergoing his internship with us since January 13, 2020 under the SDWAN, Enterprise Team.

He has undergone various trainings and completed the project as directed.

If you have any questions regarding the information provided, please don't hesitate to contact me at devkuma2@cisco.com

Yours truly,

4/17/2020

X [illegible]

Devendra Kumar
Cisco
Signed by: CiscoTrustedDevice

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706

Phone: 408 526-4000
Fax: 408 526-4100
www.cisco.com

Acknowledgement

We would like to thank CHRIST (Deemed to be University)Vice Chancellor, **Dr. Rev. Fr. Abraham V M**, Pro Vice Chancellor,**Dr. Rev. Fr. Joseph C C**, Director of School of Engineering and Technology, **Dr. Rev. Fr. Benny Thomas** and the Dean **Dr. Iven Jose** for their kind patronage.

We would like to express my sincere gratitude and appreciation to the Head of Department of Department of Computer Science and Engineering, School of Engineering and Technology **Dr. K Balachandran**, for giving me this opportunity to take up this project.

We also extremely grateful to my guide, **Chinthakunta Manjunath**, who has supported and helped to carry out the project. His constant monitoring and encouragement helped me keep up to the project schedule.

We also extremely grateful to my co-guide, **Gururajan Sundar Rao** , who has supported and helped to carry out the project. His constant monitoring and encouragement helped me keep up to the project schedule.

Declaration

We, Hereby declare that the Project titled "**Network Dashboard for SDWAN**" is a record of original project work undertaken by us for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering**. We have completed this study under the supervision of **Chinthakunta Manjunath**, Department of Computer Science and Engineering and **Gururajan Sundar Rao** , Cisco.

We also declare that this project report has not been submitted for the award of any degree, diploma, associate ship, fellowship or other title anywhere else. It has not been sent for any publication or presentation purpose.

Place: School of Engineering and Technology, CHRIST (Deemed to be University), Bangalore
Date:

Name	**Register Number**	**Signature**
Deepanshu Goyal	1660326	

Abstract

Moving applications to the cloud requires faster, more reliable connectivity. IoT demands even more performance as connected consumer endpoints multiply, taxing bandwidth and exposing sensitive networks to threats and vulnerabilities. Meanwhile, workforce is becoming more mobile, demanding optimal performance wherever it goes.

It's an arduous landscape for a business to manage, but it can be navigated with confidence when using Cisco SD-WAN. Combining software-defined efficiency with the Cisco platforms you have come to trust, Cisco SD-WAN provides unparalleled visibility across your WAN, optimal connectivity for end users, and the most comprehensive security platform to protect your network.

Through the Cisco SD-WAN vManage console, you can quickly establish an SD-WAN overlay fabric to connect data centers, branches, campuses, and colocation facilities to improve network speed, security, and efficiency. After setting templates and policies, Cisco SD-WAN analytics identifies connectivity and contextual issues to determine optimal paths for users to get to their destination, regardless of their connectivity.

Whether hosted in the cloud or on premises, Cisco vBond and vSmart orchestration and controller platforms authenticate and provision network infrastructure, verifying that the devices connecting to your SD-WAN are authorized. Once connected, SD -WAN platforms find the best path to bring users closer to the applications they need, managing overlay routing efficiency, adjusting in real time to reflect policy updates, and handling key exchanges in Cisco's full-mesh, encrypted delivery.

Contents

LIST OF FIGURES

GLOSSARY

.

Item	Description
BI	**B**usiness **I**ntelligence
CI/CD	**C**ontinuous **I**ntegration and **C**ontinuous **D**elivery
CLI	**C**ommand **L**ine **I**nterface
CSS	**C**ascading **S**tyle **S**heets
CT	**C**orporate **T**echnology
DevOps	**D**evelopment and **O**perations
DI	**D**ependency **I**njection
DVCS	**D**istributed **V**ersion **C**ontrol **S**ystems
JSON	**J**avaScript **O**bject **N**otation **S**ystems
KPI	**K**ey **P**erformance **I**ndicator
MEAN	**M**ongoDB **E**xpressJS **A**ngular **N**odeJS
MDD	**M**odel **D**riven **D**evelopment
NPM	**N**ode **P**ackage **M**anager
REST	**RE**presentational **S**tate **T**ransfer
SDLC	**S**oftware **D**evelopment **L**ife **C**ycle
CISCO	CISCO **S**ystems **I**ndia
UI	**U**ser **I**nterface

Chapter 1

INTRODUCTION

A software-defined wide-area network (SD-WAN), is a network that is abstracted from its hardware, creating a virtualized network overlay. Operators can remotely manage and quickly scale this overlay, which can span over large geographical distances. It is an application of software-defined networking (SDN).

An SD-WAN can connect several branch locations to a central hub office or cover multiple locations in a large campus such as a university campus.

Because SD-WAN is abstract from hardware, it is more flexible and available than a standard WAN. It relies on four central components: Edge connectivity abstraction, WAN virtualization, Centralized management, Elastic traffic management

Because of its virtualized architecture, SD-WAN doesn't require specific hardware for specialized network functions. Instead, the infrastructure is made of commercial off-the-shelf (COTS) equipment, also known as whiteboxes.

Enterprises can deploy SD-WAN in a DIY manner, where the business owns the network and equipment and takes full responsibility for the network operation and upkeep. In turn, enterprises can use a managed service provider, who owns all network equipment and maintains some control over the network, and takes the brunt of the network management responsibility.

1.1 Problem Identification

Network performance is critical as far as an enterprise is concerned. The network performance can be segregated into network speed and network reliability. Both of them are key performance parameters for an enterprise network.

If an enterprise network becomes unstable with higher downtime, then it will impact the overall performance of an enterprise network. Moreover, in case of an unscheduled outage, the break-fix solution might include replacement of legacy devices or failed devices. This costs both, time and resources. It impacts productivity as well.

WAN outages have been one of the top contributors that negatively impact the productivity of enterprise networks.

Today, the majority of the organizations have connected their enterprise networks to the cloud and often to multiple clouds. However, multi-cloud architectures pose certain challenges for the enterprise network. It will be a challenge to manage the different providers and apply an integrated security standard to all the providers. At the same time, it will be difficult to strike a proper balance between on and off-premises environments.

This includes the challenge of deriving a perfect model that can connect on-premise datacenters to the cloud. An enterprise network can deliver a better performance with reliability if the on-premise environment and off-premise environment is perfectly balanced. This should be defined by a proper cloud strategy of an organization.

1.2 Problem Formulation

To create user-friendly websites which relay information effectively to a target audience with the help of visual cues, easy navigation, and effective user interaction. To develop websites which are responsive with mobile, tablet, laptop and desktop. To create a website which can be focussed to attract the target audience.

1.3 Problem Statement & Objectives

Transport independence: Guaranteeing zero network downtime, Cisco SD-WAN automates application flexibility over multiple connections, such as the Internet, MPLS, and wireless 4G LTE.

Network services: Rich networking and security services are delivered with a few simple clicks. WAN optimization, cloud security, firewalling, IPS, and URL filtering can be deployed wherever needed across the SD-WAN fabric from a single location.

Endpoint flexibility: Cisco SD-WAN can simplify connectivity across branches, campuses, data centers, or cloud environments, extending the SD-WAN fabric wherever you need it to go. Whether physical or virtual, a variety of Cisco SD-WAN platforms gives you unparalleled choice, ensuring that your specific business needs are met.

As the world's leading networking company, Cisco defined the standard for routing. As the largest enterprise cybersecurity provider, Cisco covers thousands of customers with end-to-end protection.

By choosing Cisco SD-WAN, you gain the ability to manage certified trustworthy platforms while instantly deploying the right security in the right place, all from a single dashboard. With a few clicks in the Cisco vManage console, you can instantly protect your entire network, reducing risk while ensuring business compliance, continuity, and success.

Cisco SD-WAN can transform your Cisco routers into advanced, multilayered security devices with an application-aware enterprise firewall, IPS, URL filter, and continuous DNS monitoring. As a result, end users–whether in the data center, branch, campus, or a remote location–can enjoy protection from a multitude of security threats. In addition, Cisco SD-WAN can segment network traffic end to end, protecting your business against data exfiltration and insider threats.

1.4 Limitations

Software is more susceptible to errors, bugs and glitches, which is why, for years, WAN vendors preferred to hardcode functions into hardware for more stability. By moving to SD-WAN, customers will gain tremendous operational flexibility, but that comes at the expense of the peculiarities and limitations of software.

Troubleshooting - Because SD-WAN includes both an underlay and overlay in the WAN service – and possibly from two different vendors – troubleshooting becomes more complex. WAN issues then require two different places to investigate before zeroing in on the problem.

Technical skills - SD-WAN completely upends the WAN status quo and will require a lot of training for IT teams because of the new approach to WAN management. IT pros can't simply drop an SD-WAN in and let the teams manage it. IT departments must also ensure they allocate enough money for their training budgets.

Security - WANs represent significant exposures because they are outward-facing and rely on transport outside the firewall. Deploying any new WAN technology can introduce unintended security issues. Additionally, vulnerabilities could now exist across both WAN layers.

Complexity - Typically, WAN connections are a single service from one company. SD-WAN adds an extra component on top, bringing an overlay to the existing WAN connection – the underlay – creating complexity.

Chapter 2

LITERATURE SURVEY AND REVIEW

The point of enumerating all the portions of modern enterprise IT communications that SD-WANs don't manage, is not to diminish the benefits of SD-WANs. Rather, it is to highlight the growth and criticality of the external domain. The digital business, brand reputation, employee productivity and revenue are all critically dependent on external providers, networks and services. The fact that neiter control those resources doesn't let you off the hook in terms of the business outcomes.

In the absence of control, visibility is key. That's why 5 of the top 6 U.S. banks, 8 of the top 10 global software companies, 18 of the top 20 SaaS and 50+ of the Fortune 500 use Network Intelligence delivered by ThousandEyes to see, understand and optimize connected experiences for their customer-facing apps, as well as their WAN modernization efforts and cloud adoption strategies. Apropos to SD-WAN, we see more enterprises embracing a cloud readiness lifecycle and deploying ThousandEyes visibility ahead of time to prepare for SD-WAN rollouts to gain an understanding of how the Internet will behave and how cloud resources will perform from their distributed locations. From there, they're ready to operate successfully in both internal and external dimensions of their enterprise IT communications. If you're ready to learn more about how Network Intelligence complements and helps the success of SD-WANs, request a demo and we'll walk you through how to get the overall visibility you need to modernize your WAN.

2.1 Literature Collection & Segregation

In the paper ”Achieving high utilization with software-driven WAN” the author talks about Cloud computing data centers are becoming increasingly popular for the provisioning of computing resources. In the past, most of the research works focused on the effective use of the computational and storage resources by employing the Virtualization technology. Network automation and virtualization of data center LAN and WAN were not the primary focus. Recently, a key emerging trend in Cloud computing is that the core systems infrastructure, including compute resources, storage and networking, is increasingly becoming Software-Defined. In particular, instead of being limited by the physical infrastructure, applications and platforms will be able to specify their fine-grained needs, thus precisely defining the virtual environment in which they wish to run. Software-Defined Networking (SDN) plays an important role in paving the way for effectively virtualizing and managing the network resources in an on demand manner. Still, many research challenges remain: how to achieve network Quality of Service (QoS), optimal load balancing, scalability, and security. Hence, it is the main objective of this article to survey the current research work and describes the ongoing efforts to address these challenging issues.

Software-Defined Networking (SDN) is recently an emerging technique that paves the way for virtualizing the network resources in an on demand manner. It provides an abstraction of the underlying network to the applications residing in upper layers. Conventionally, the network devices such as switches and routers have control plane, management plane and data plane whereas in SDN, the logic of control and data plane is decoupled separately. The control plane logic is implemented as a software component that is residing in a server and data plane is located in network devices. The decoupling of control and data plane logic has transformed the network resources into programmable, automation and network control, highly scalable and flexible networks based on the business needs. Moreover, SDN replaces the functionality of networking devices as just forwarding devices. The intelligence of where and how to make 2013 Fifth International Conference on Advanced Computing (ICoAC) forwarding is residing in control plane. The control plane logic is implemented in the software called controller. OpenFlow is the protocol for communicating the controller with network devices. Some of the popular SDN controllers in the market and research are Floodlight, Beacon, NOX, and OpenDayLight. The SDN architecture used in current networking world is shown in Figure 1. The SDN application 1 to N represents the features such as Quality of Service (QoS), Load Balancing (LB), Firewall (FW) and etc. that is deployed on top of SDN controllers. The Controller receives the packet and forward to OpenFlow based switches for example OpenVSwitch.

2.2 Critical Review of Literature

Cloud resources such as compute, storage and network become the worthwhile infrastructure for computation, data storage and hosting network based applications. SoftwareDefined Networking (SDN) solves the issues in the conventional networking and virtualizes the network resources in an on demand manner to maximize the utilization by effectively using the network resources and satisfying the user application constraints. In this paper, we surveyed the state of the art in Software-Defined Networking (SDN) research in four areas: Network Quality of Service (QoS), Load Balancing, Scalability and Security. From the literature survey, we have identified that, there is no common architecture or solution to address all the four issues that should be addressed in the context of Software-Defined Networking (SDN)

Chapter 3

ACTUAL WORK

Before starting with the actual work on the project, there are a lot of things that need to be kept in mind so as to get better results from the projects. These things include distributing the responsibilities among the team, the tools that will be used, the methodology that will be deployed in the making of this project, etc. It was important to understand the working of website and the need of customers. Also before beginning the projects, a detailed study of Bootstrap was required so as to make use of it effectively. This chapter is divided into three parts which explains how the projects were undertaken. They are as follows:

- Methodology for the Study
- Implementation Details
- Prototype used for all the projects

3.1 Methodology for Web Development

There are various methodologies that can be used for web development, these are based on the idea to leverage the buisness. Aim while making a website is to understand and analyze the buisness requirements, and develop it by keeping these in minds.People use waterfall model, agile model and vaious other models to create a website.

Methodologies used by the industry is agile model along with kanban to develop websites.

3.1.1 Agile Methodology

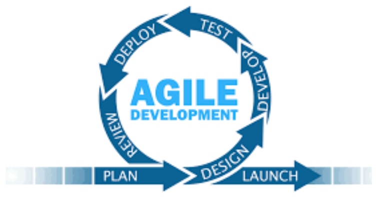


FIGURE 3.1: Agile Methodology

There is a drastic change in the era of software development. Keeping up with these current trends, technologies and methodologies is utmost important. A web development agency generally selects a process which is easier, faster and efficient

There is no more use of traditonal methodologies in delivering products. These old methodologies like waterfall are replaced by Agile Methodology, which focuses on continous delivery of websites as per requirements specified by the client .

This methodology is a very flexible method, is a time- focused philosophy that allows creating projects incrementally, by dividing it into small chunks.The benefit of usinf such methodology is that changes can be made at any step during development. Each sprint will involve a daily review hence keeping the client as well as the team updated about the project.

The Agile methodology was incorporated in 2001 in Utah by a group of software developers. It started with a trial and error method with frequent setbacks. But as time evolved, large experiments done with this methodology flourished.

Agile method or methodology has evolved into a life cycle which has benefited the web development agencies and the end users. It turns out to be the most efficient and reliable source of working style. The key was to do smarter and less work and deliver it sooner.

Let's observe the lifecycle of the Agile methodology:

1. Understanding the client's requirement
2. Sprint Planning
3. Designing the product
4. Testing

Benefits of using Agile Methodology

1. **Consumes Less time** : As frequent feedbacks are received from the client, the bugs are fixed quickly, thereby completing the project on time.
2. **Client satisfaction** : Demonstrating each sprint of work to the client gives huge flexibility and visibility in the progress of their work.
3. **Reduced Risks** : Agile methodology eliminates the chances of complete project failure. It is a highly collaborative approach to work with the end user.
4. **Increases business value** : The web development takes special effort to understands what is important to the client's business and deliver in the most valuable order.
5. **Maintains transparency** : Direct communication with the client creates a high level of transparency and makes work much simpler.
6. **Adaptive approach** : There is a regular adaption to changing circumstances and late changes suggested by the client are also proved successful.

3.1.2 Kanban Methodology

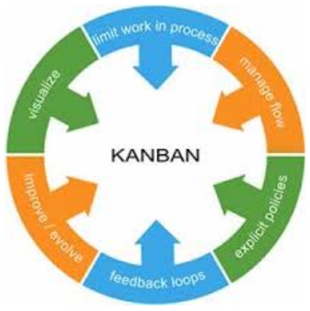


FIGURE 3.2: Kanban Methodology

Kanban is the most widely preferred agile software development framework. Basically, this excellent development is suitable for those development projects that are constantly altering or extremely developing requirements.The main idea of Kanban is workflow visualization. It consists of creating a physical panel (Kanban board) on which you can visually mark progress. A kanban board may be shared by multiple teams or individuals.

Kanban is based on 3 major principles:

1. Visualizing what one do in a day(workflow) -
2. Limiting the amount of work in progress - i.e. not commiting to too much work at once
3. Once one part is completed, the next highest thing comes into play.

Benefits of using a Kanban Methodology:

1. Shorter cycle times cab deliver features faster
2. Kanban is ideal if there are changes in priorites.
3. Easy to make changes.
4. Requires fewer organization / room set-up changes to get started
5. Reducing waste and removing activities that don't add value to the team/department/organization
6. Rapid feedback loops improve the chances of more motivated, empowered and higher-performing team members

3.2 Modeling, Analysis and Design

As mentioned earlier Front-end designing was taken up, as first priority. The framework used was Angular on IDE Visual Studio Code. Basic layout of a project was created using Angular CLI (Command Line Interface) by administering the right command.

3.2.1 File Structure

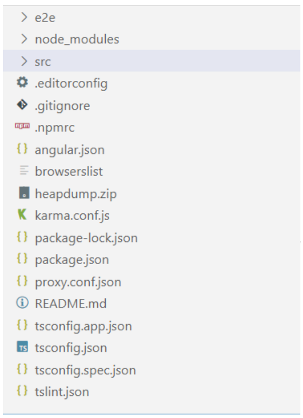
A typical Angular project file structure is given in the figure below:

From the above figure,

1. src – Location of source files.

2. dist – stores all the built files.

3. package.json - lists all the node dependencies that are available to the project.



```
> e2e
> node_modules
> src
.editorconfig
.gitignore
.npmrc
angular.json
browserslist
heapdump.zip
karma.conf.js
package-lock.json
package.json
proxy.conf.json
README.md
tsconfig.app.json
tsconfig.json
tsconfig.spec.json
tslint.json
```

FIGURE 3.3: File Structure

4. node-modules - lists all the npm (Node Package Manager) packages that are used.

5. package-lock.json - lists the versions of all the packages that are installed.

6. angular.json - lists the default configurations of the Angular CLI.

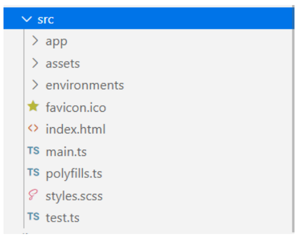
7. tsconfig.json - lists the default configurations of Typescript.

8. tslint.json - lists the default configurations of Lint.

9. .editorconfig - lists the default configurations of the Editor.

10. .gitignore - lists all the files that has to be ignored by git.

On expanding src, another set of files are unveiled. The figure below shows the following:



```
v src
  > app
  > assets
  > environments
  favicon.ico
  index.html
  main.ts
  polyfills.ts
  styles.scss
  test.ts
```

FIGURE 3.4: Files under src

From the above figure,

• src/app - stores the source files, data and logic.

• src/assets - stores the images used across the application.

• src/environments - enlists the build configuration details at certain target environments. By default, there are none.

• favicon.ico - contains the icon the bookmark must hold.

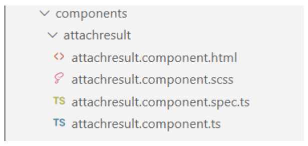
• src/index.html - the HTML page that one observes on visiting the site.

• src/main.ts - main entry point in the application from where compilation begins.

• src/style.css - describes all the global styles that is applicable across the entire application.

• polyfills.ts - stores polyfills script for browser support.

• tests.ts - entry point to all unit tests undertaken for the application.



```
components
  attachresult
    attachresult.component.html
    attachresult.component.scss
    attachresult.component.spec.ts
    attachresult.component.ts
```

FIGURE 3.5: Files under components

This is a typical Angular File structure that is automatically generated by the CLI, on giving the command. Further below is the structure that the file has taken up, in the course of the development. On expanding src/app, the following window unveils. From the above figure, it is clear that there are 5 other folders within src/app. All source files written during the project are categorized into these.

• Components – every small feature/functionality of a project is considered a component. This folder houses all the components of the application.

• Interfaces – As the names suggest, this folder encases all the interfaces used.

• Routing – On meeting the necessary conditions, the page redirects to another page. This is called Routing. The folder enlists all the routes administered in the application.

• Service – A service is a small bit of code that can be reused anywhere and wired using Dependency Injection (DI). The folder contains all the services implemented in the project.

• TRIAL – Before incorporating into the main work-flow, all pieces of code were tested and implemented in this folder first.

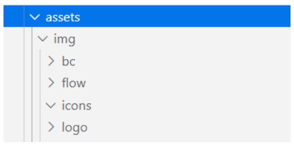


FIGURE 3.6: Files under assets

On expanding src/assets, the following window unveils.

From the above figure, it is clear that app/assets contains all the images that are used throughout the application.

On visiting the site, the HTML page that was served included a navigation bar, a navigation tool bar and the Project grid.

The Navigation bar housed the Company's logo, a Home button and a search functionality. The second navigation bar included a Filter option and the Paginator service that helped navigate through the various projects.

As mentioned earlier the project grid encased 8 project tiles, each showing project-specific information.

Each of these features were implemented through Components.

3.2.2 Component Structure

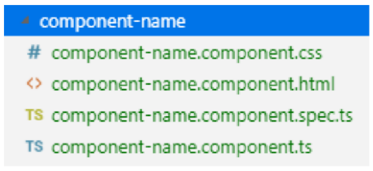

component-name
component-name.component.css
component-name.component.html
component-name.component.spec.ts
component-name.component.ts

FIGURE 3.7: Component Structure

A component can be generated in an Angular project by simply using the Angular CLI and the right command. A typical component has the following structure:

• .component.css - A CSS (Cascading Style Sheets) style sheet that styles only that component.

• .component.html - A html page that contains the basic markup of the component.

• .component.spec.ts - To initiate unit tests for the component.

• .component.ts - Stores the data and governing logic that controls the functionality of the component.

3.3 Implementation Details

The vDoctor tool can be the one place for all the necessary data a TAC engineer requires for trouble shooting resources. Details about vManage, vSmart, vBond and Edge devices are available together. It saves the time required to go to each and every vManage, revisit through pages again and again by displaying the information about controllers in one page and all the devices in another page in GUI tool. vDoctor is a dedicated tool for monitoring the customer's network

The vDoctor tool has only four pages. In first page – user has to enter valid credentials to login to vDoctor In second page – user has to enter vManage IP address and the credentials to get the parameters of desired network In third page – All the necessary data of controllers (vManage, vSmart and vBond) is present. In fourth page – All the necessary data of edge devices (vEdge or cEdge) is present. The detail information about each page is discussed in next section. High level design of vDoctor is shown in Fig:

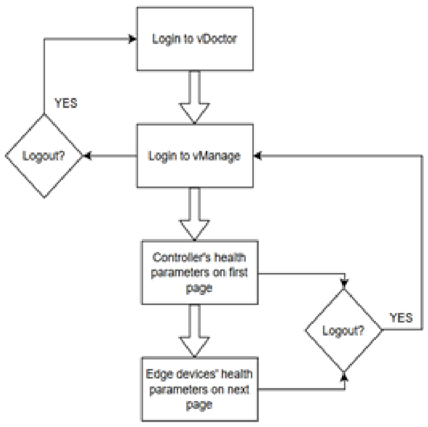

Login to vDoctor
YES
Logout?
Login to vManage
Controller's health parameters on first page
Logout?
YES
Edge devices' health parameters on next page

FIGURE 3.8: High Level Working of vDoctor

3.3.1 Project Low Level insights

In this section, details of each card is given one by one from login page to the last page. Also, a low level design shows the detailed flow of the tool as below:

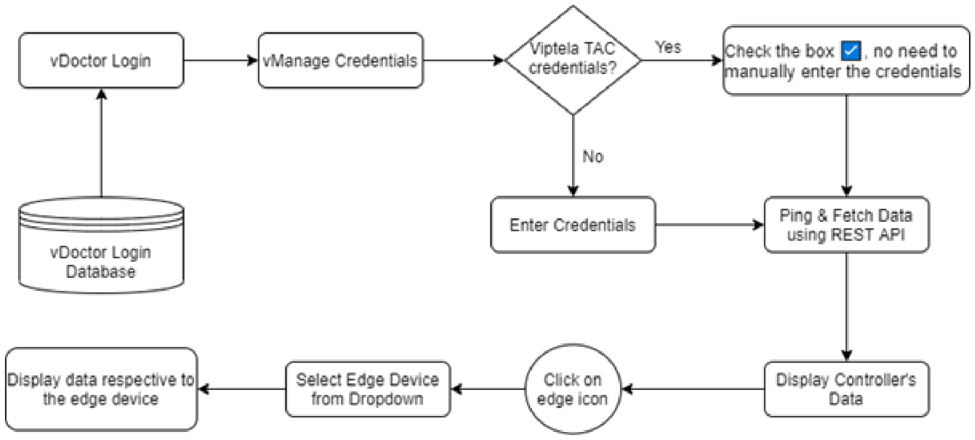

vDoctor Login
vManage Credentials
Viptela TAC credentials?
Yes
Check the box ✓, no need to manually enter the credentials
No
vDoctor Login Database
Enter Credentials
Ping & Fetch Data using REST API
Display data respective to the edge device
Select Edge Device from Dropdown
Click on edge icon
Display Controller's Data

FIGURE 3.9: Low Level Working of vDoctor

3.3.1.1 vDoctor Login

vDoctor Login page is required for successful authorization. Field 'Username' is usually CEC ID of Cisco Engineers. To get the credentials user has to contact vDoctor@help.com. vDoctor administrators can login using a default credentials also. This default credential is confidential and only vDoctor administrators can access it. Back End – A 'sql3' database is created to store the credentials. Only the credentials stored in the database can access the tool.

3.3.1.2 vManage Login

After user login to vDoctor, Details about vManage has to be entered. The desired network's vManage IP address, for which user want to check the resources, has to be provided. The user has two options to proceed- Firstly use the credentials provided, or secondly use the default 'viptelatac' confidential credentials. To access the resources using 'viptelatac' credentials, check the box 'Login with Viptela TAC' and leave the other fields (Username and Password) empty. To access using other credentials, provide all the fields. j-security-check method is used for authorization. Session ID is created while login. This method of authorization is compatible with latest vManage release

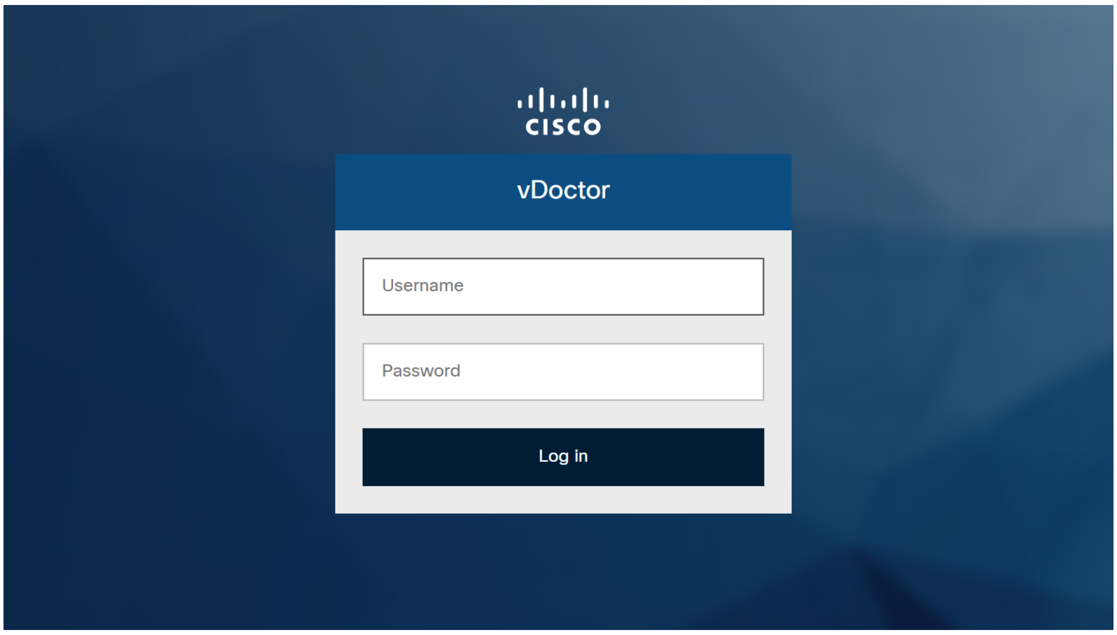


FIGURE 3.10: vDoctor Login Page

and older. Although, in later updates of vDoctor, the token generation method can be incorporated for login authorization. The logout button is used to logout from vDoctor.

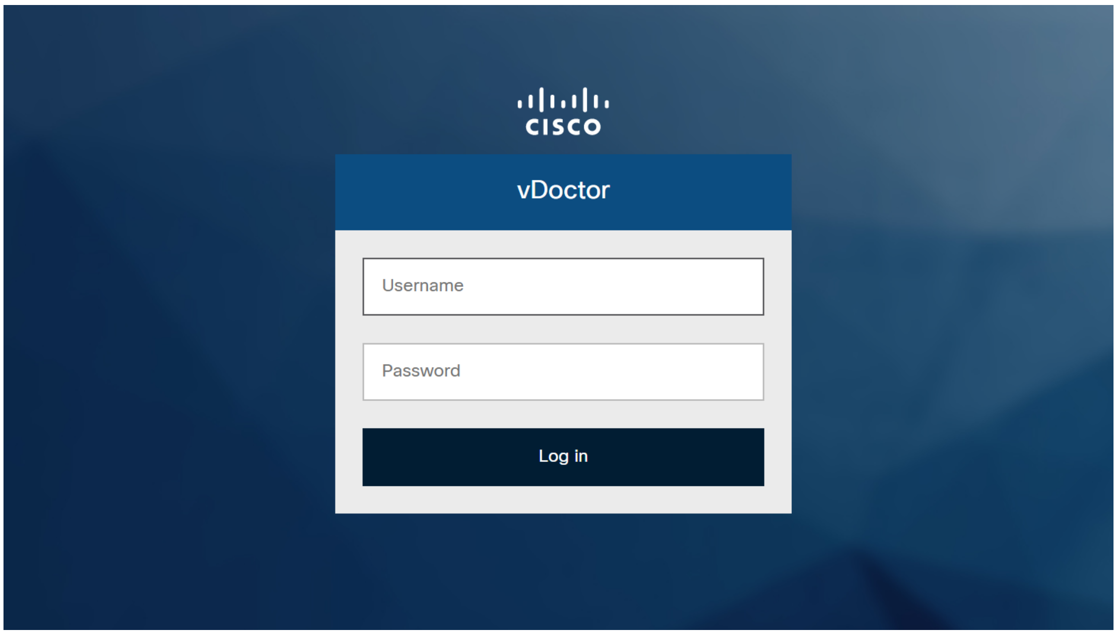
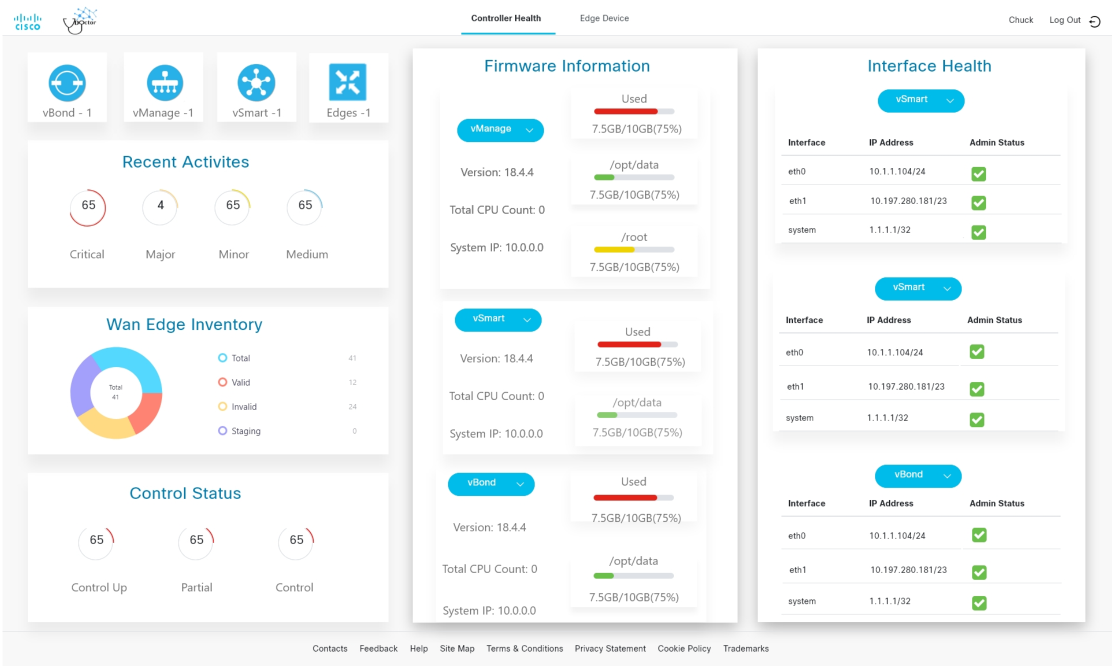


FIGURE 3.11: VManage Login Page

3.3.1.3 Home Page

Home page of vDoctor displays information related to controllers of network i.e. vManage, vSmart and vBond. There are six cards in this page.

Refresh button in each card can be used to refresh the data of the particular card only. User need not reload the whole page. However, refresh button in network summary refreshes interface health and firmware information as well. It is required because number of devices has to be updated in the latter cards.



Recent Activites

Critical	Major	Minor	Medium
65	4	65	65

Wan Edge Inventory

Total 41

Total	41
Valid	12
Invalid	24
Staging	0

Control Status

Control Up	Partial	Control
65	65	65

Firmware Information

vManage
Version: 18.4.4
Total CPU Count: 0
System IP: 10.0.0.0
Used 7.5GB/10GB(75%)
/opt/data 7.5GB/10GB(75%)
/root 7.5GB/10GB(75%)

vSmart
Version: 18.4.4
Total CPU Count: 0
System IP: 10.0.0.0
Used 7.5GB/10GB(75%)
/opt/data 7.5GB/10GB(75%)

vBond
Version: 18.4.4
Total CPU Count: 0
System IP: 10.0.0.0
Used 7.5GB/10GB(75%)
/opt/data 7.5GB/10GB(75%)

Interface Health

vSmart

Interface	IP Address	Admin Status
eth0	10.1.1.104/24	✓
eth1	10.197.280.181/23	✓
system	1.1.1.1/32	✓

vSmart

Interface	IP Address	Admin Status
eth0	10.1.1.104/24	✓
eth1	10.197.280.181/23	✓
system	1.1.1.1/32	✓

vBond

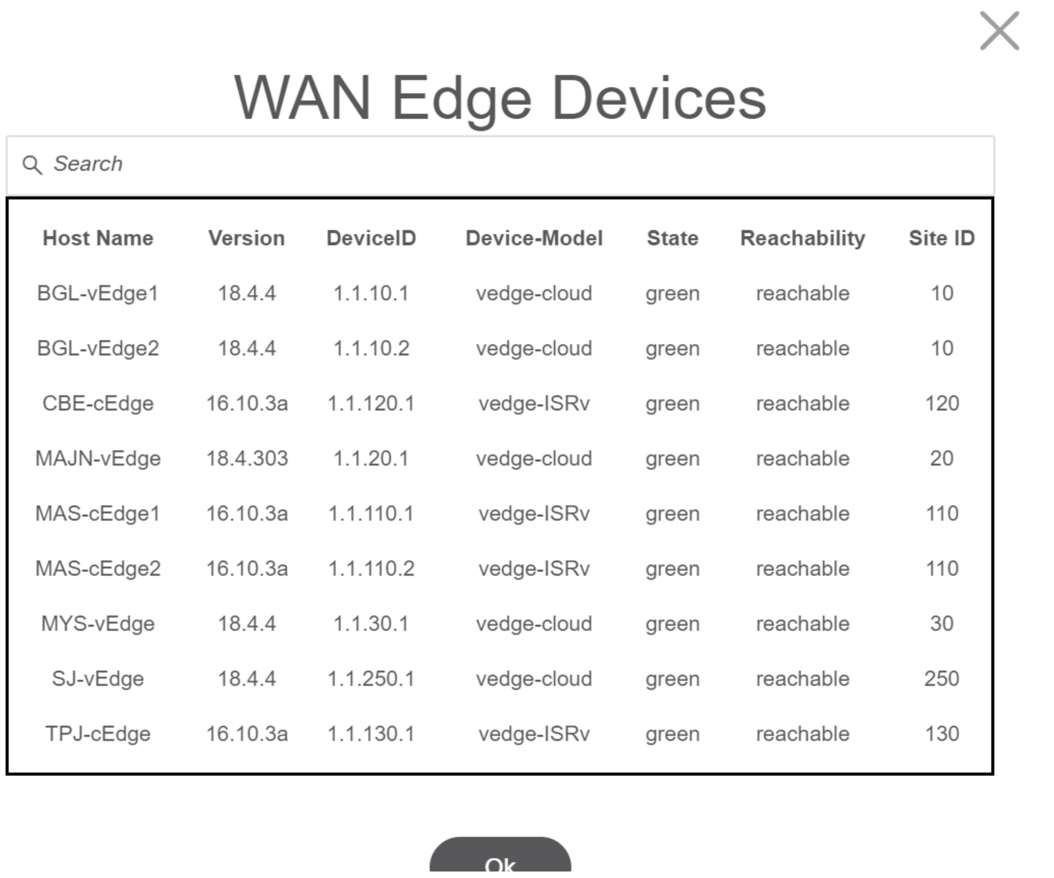
Interface	IP Address	Admin Status
eth0	10.1.1.104/24	✓
eth1	10.197.280.181/23	✓
system	1.1.1.1/32	✓

FIGURE 3.12: Homepage — Controller Health Page

3.3.1.4 Network Summary

In this card, total number of devices are categorized. The ‘Edges’ block is clickable. If user clicks on ‘Edges’ label, a table with basic details about edges (vEdge and cEdge) will pop up as shown in figure below. The table contains – Host name, version, system IP, device model, reachability and site ID. If the ‘Edges’ icon is clicked, it will load the next page with more details about the Edge devices.

API call used to get this data is: https://vmanage_ip_address $/dataservice/device$


WAN Edge Devices

Search

Host Name	Version	DeviceID	Device-Model	State	Reachability	Site ID
BGL-vEdge1	18.4.4	1.1.10.1	vedge-cloud	green	reachable	10
BGL-vEdge2	18.4.4	1.1.10.2	vedge-cloud	green	reachable	10
CBE-cEdge	16.10.3a	1.1.120.1	vedge-ISRv	green	reachable	120
MAJN-vEdge	18.4.303	1.1.20.1	vedge-cloud	green	reachable	20
MAS-cEdge1	16.10.3a	1.1.110.1	vedge-ISRv	green	reachable	110
MAS-cEdge2	16.10.3a	1.1.110.2	vedge-ISRv	green	reachable	110
MYS-vEdge	18.4.4	1.1.30.1	vedge-cloud	green	reachable	30
SJ-vEdge	18.4.4	1.1.250.1	vedge-cloud	green	reachable	250
TPJ-cEdge	16.10.3a	1.1.130.1	vedge-ISRv	green	reachable	130

FIGURE 3.13: WAN Edge Device Modal

3.3.1.5 Recent Activities

This card shows number of alarms categorized based on severity, e.g. Critical, major, minor and medium. Critical - Serious events that impair or shut down the operation of an overlay network function. Major - Serious events that affect, but do not shut down, the operational of a network function. Medium - Events that might impair the performance of a network function. Minor - Events that might diminish the performance of a network function. The alarms are shown for last 7 days here.

Note—In later updates of this tool, option to customize the time user want to check alarms can be added.

When user click on any of the alarms icon, a table with list of all the specific severity alarms is appeared. The table contains – Date Time, Impacted entities and message as shown in figure below. Date Time – Time when the alarm was raised, expressed in Indian Standard Time (IST) Impacted entities – Information about the affected devices.

It is different for different alarms. Host name, system IP, site id are typical values present in this column. Message – Short description about the alarm.

API call used to get this information is:

https://vmanage_ip_address $/dataservice/alarms?query =$

To generate the query:

• Go to https://vmanage_ip_address $/apidocs$

• Go to monitoring – Alarm details

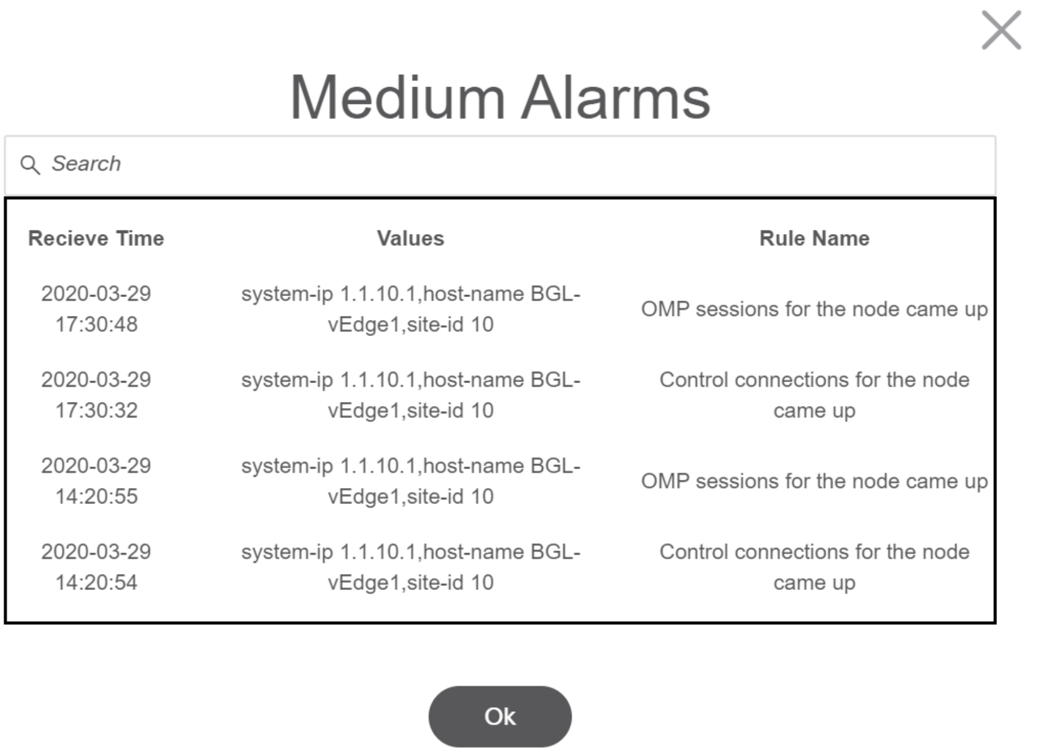
• Go to "GET" query (/alarms)

Query to get 10,000 alarms for last 6 hours is:

```
{
    "query" : {
        \condition" : \AND" ,
         \rules" : [
        {
            \value" : [
                \6"
            ] ,
                \field" :  \entry_time" ,
                \type" : \date" ,
                \operator" : \last_n_hours"
        }
        ]
    } ,
    \size" : 10000
}
```

3.3.1.6 WAN Edge Inventory

The WAN Edge inventory card provides four counts. To get the details of these four counts, user has to click on the label. The details in each label are: Total – Basic



Recieve Time	Values	Rule Name
2020-03-29 17:30:48	system-ip 1.1.10.1,host-name BGL-vEdge1,site-id 10	OMP sessions for the node came up
2020-03-29 17:30:32	system-ip 1.1.10.1,host-name BGL-vEdge1,site-id 10	Control connections for the node came up
2020-03-29 14:20:55	system-ip 1.1.10.1,host-name BGL-vEdge1,site-id 10	OMP sessions for the node came up
2020-03-29 14:20:54	system-ip 1.1.10.1,host-name BGL-vEdge1,site-id 10	Control connections for the node came up

FIGURE 3.14: Recent Activities Alarms

details of WAN Edge routers whose authorized serial number has been uploaded on the vManage server. When clicked on ‘total’, a table with details – host name, system IP, chassis number, device model and validity will appear as shown in figure below. Valid – Devices with valid certificate. Details like host name, system IP, device model and reach ability will appear in pop up. Invalid – Devices with invalid certificate. Details shown are device model, chassis number and serial number. Staging – Edge routers that are in staging stage. Details shown are device model, chassis number and serial number.

API call used to get this data is:

https://vmanage_ip_address $/dataservice/certificate/vedge/list$

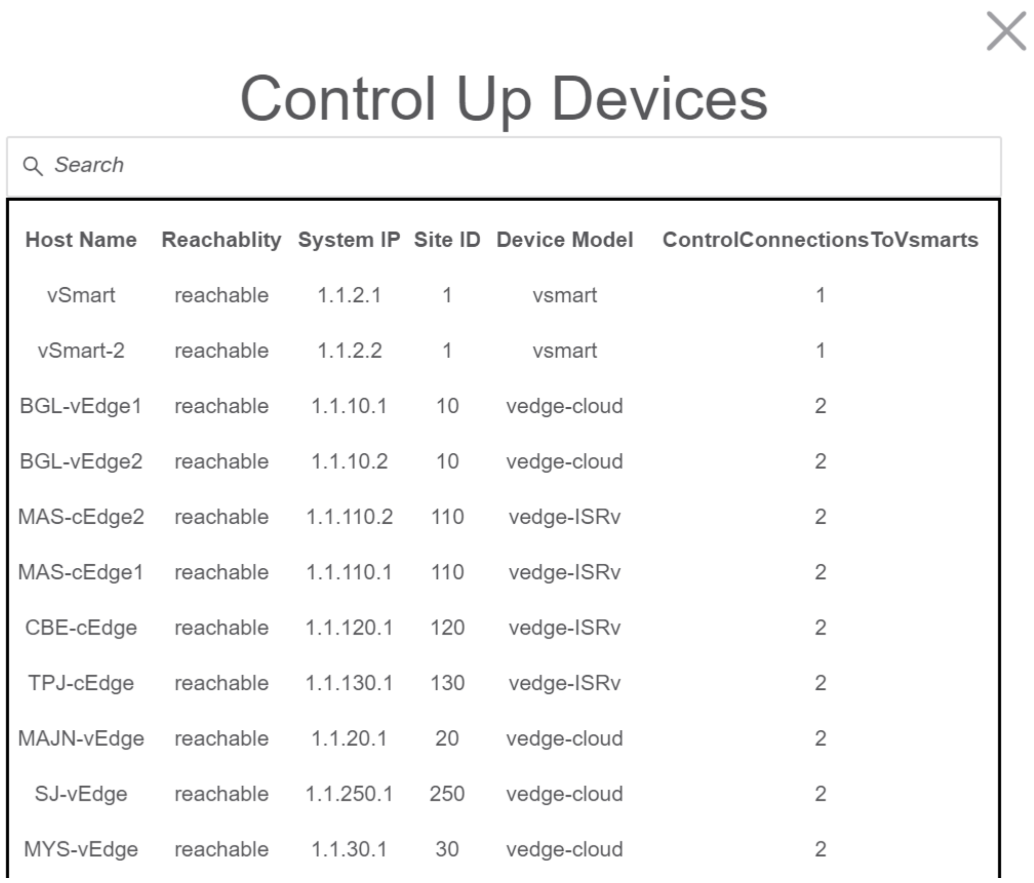
3.3.1.7 Control Status

The Control Status pane displays whether vSmart and WAN Edge devices are connected to the required number of vSmart controllers. The Control Status pane shows three counts: Control Up—Total number of devices with the required number of operational control plane connections to a vSmart controller. Partial—Total number of devices with some, but not all, operational control plane connections to vSmart controllers. Control Down—Total number of devices with no control plane connection to a vSmart controller. When click on any of the icon, it gives details about the devices such as Host name, reach-ability, system IP, Site id, Device model, control connections to vSmart as shown in figure below.

API call used to get the details are:

https://vmanage_ip_address $/device/control/networksummary?state=$

Here, state is either up or partial or down.



Control Up Devices

Search

Host Name	Reachablity	System IP	Site ID	Device Model	ControlConnectionsToVsmarts
vSmart	reachable	1.1.2.1	1	vsmart	1
vSmart-2	reachable	1.1.2.2	1	vsmart	1
BGL-vEdge1	reachable	1.1.10.1	10	vedge-cloud	2
BGL-vEdge2	reachable	1.1.10.2	10	vedge-cloud	2
MAS-cEdge2	reachable	1.1.110.2	110	vedge-ISRv	2
MAS-cEdge1	reachable	1.1.110.1	110	vedge-ISRv	2
CBE-cEdge	reachable	1.1.120.1	120	vedge-ISRv	2
TPJ-cEdge	reachable	1.1.130.1	130	vedge-ISRv	2
MAJN-vEdge	reachable	1.1.20.1	20	vedge-cloud	2
SJ-vEdge	reachable	1.1.250.1	250	vedge-cloud	2
MYS-vEdge	reachable	1.1.30.1	30	vedge-cloud	2

FIGURE 3.15: Control Up Devices modal

3.3.1.8 Firmware information

This section contains the firmware details of the controllers such as software version, total CPU count and memory allocation. These details are not directly present in vManage tool and user has to do SSH to device in vManage. This card is divided into three sections each for vManage, vSmart and vBond respectively. A drop down menu in each section is used to select the controllers form the list. CLI commands to get memory details from vManage is “df –kh” and to get CPU count is “lscpu”. There are three types of memories: /optdata: it is the data base memory and only vManage has this memory allocated.

API call used to get this data is:

https://vmanage_ip_address $/device/system/status?deviceId =$

At the end of URL, user has to mention the system IP of desired controller to get the details.

API call used to get this data is:

https://vmanage_ip_address $/device$

3.3.1.9 Interface Health

The Interface health card gives the IPv4 interface status of the controllers. Similar to Firmware information, this card is also divided into three sections each for vManage, vSmart and vBond. The drop down menu in each section is used to select the controllers from the list. The interface name, IP address, operator status and admin status are displayed in this card. API call used to get this data is:

https://vmanage_ip_address $/device/interface/synced?deviceId =$

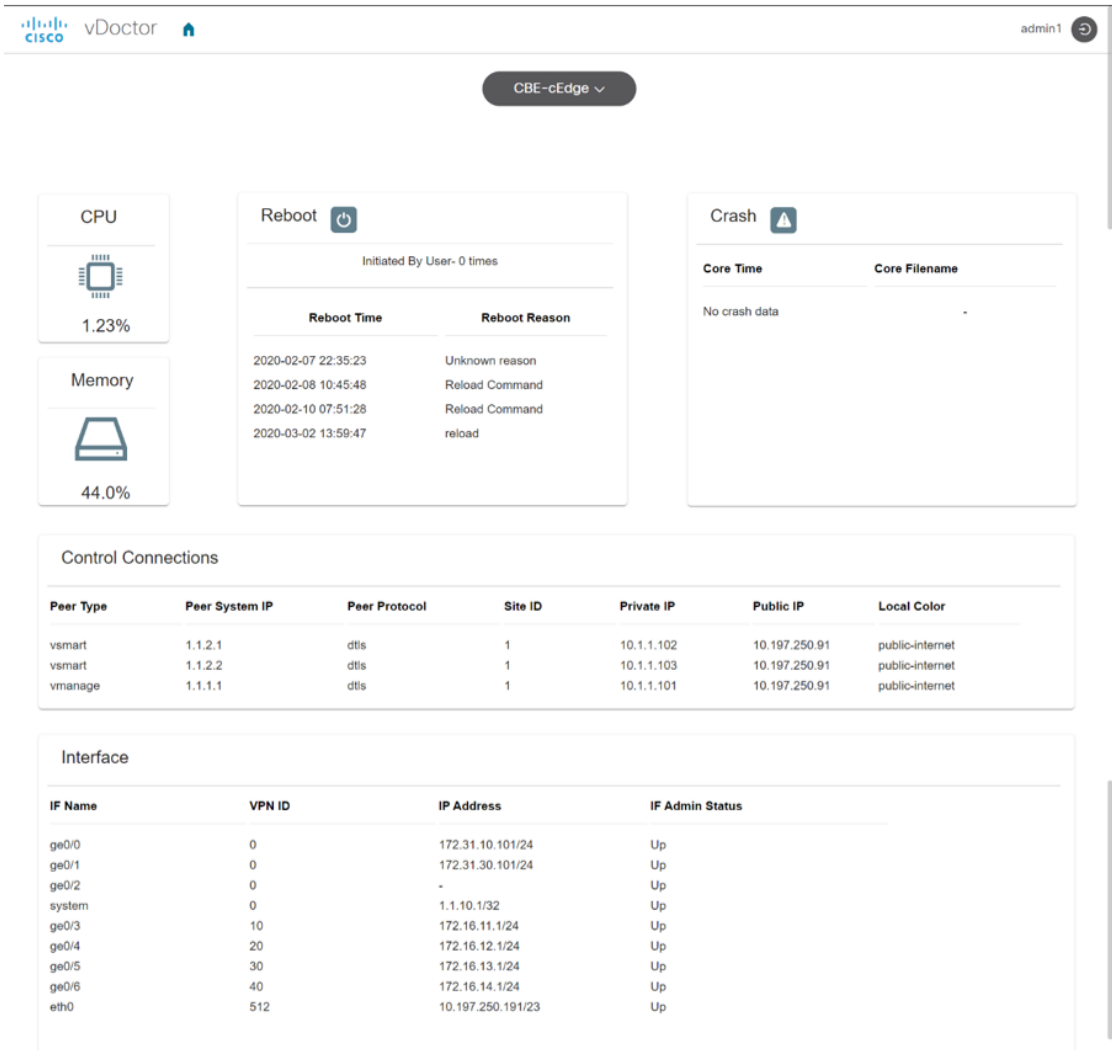
The value for device Id is the system IP of the device for which interface information need to be found.

API call used to get this data is:

https://vmanage_ip_address $/dataservice/device$

3.3.1.10 Edge details

Next Page, also the last page, contains the details about the edge devices. User has to click on ‘Edge’ icon in network summary of home page to go to the next page. The edge devices can be either vEdge or cEdge. This page has seven cards.



CPU

1.23%

Memory

44.0%

Reboot

Initiated By User- 0 times

Reboot Time	Reboot Reason
2020-02-07 22:35:23	Unknown reason
2020-02-08 10:45:48	Reload Command
2020-02-10 07:51:28	Reload Command
2020-03-02 13:59:47	reload

Crash

Core Time	Core Filename
No crash data	-

Control Connections

Peer Type	Peer System IP	Peer Protocol	Site ID	Private IP	Public IP	Local Color
vsmart	1.1.2.1	dtls	1	10.1.1.102	10.197.250.91	public-internet
vsmart	1.1.2.2	dtls	1	10.1.1.103	10.197.250.91	public-internet
vmanage	1.1.1.1	dtls	1	10.1.1.101	10.197.250.91	public-internet

Interface

IF Name	VPN ID	IP Address	IF Admin Status
ge0/0	0	172.31.10.101/24	Up
ge0/1	0	172.31.30.101/24	Up
ge0/2	0	-	Up
system	0	1.1.10.1/32	Up
ge0/3	10	172.16.11.1/24	Up
ge0/4	20	172.16.12.1/24	Up
ge0/5	30	172.16.13.1/24	Up
ge0/6	40	172.16.14.1/24	Up
eth0	512	10.197.250.191/23	Up

FIGURE 3.16: Edge Devices Page

3.3.1.11 Dropdown option

Dropdown option at the top centre is used to select the Edge device for which information is needed. This option shows the host name of the devices. Information for any one of the device will be shown by default. If the user clicks on this option, a drop-down with a list of all hostnames of edges (vEdge and cEdge) will be displayed. If any one of the edge is selected, it will load the page with all the details about the Edge devices.

3.3.1.12 Device details

Device basic details are displayed right beneath the drop down button. This contains host name, system IP, device model, site id, state, and reach ability.

API used to get this detail is:

https://vmanage_ip_address */dataservice/device*

3.3.1.13 CPU load display

CPU load is a number of processes that are being executed by CPU or waiting to be executed by CPU and measures the amount of computational work that a computer system performs. When an edge is selected from the drop-down menu, this card shows the percentage of the average load of the device over 24 hours.

API call used to get this data is:

https://vmanage_ip_address */dataservice/device/hardwarehealth/detail*

3.3.1.14 Memory Usage display

This card displays the memory usage as a percentage of available memory over the period of 24 hours.

API call used to get this data is:

https://vmanage_ip_address */dataservice/device/hardwarehealth/detail*

3.3.1.15 Reboot history

The reboot history card shows the details about number of time device reboot occurred. Reboot initiated by user is not considered important for troubleshooting purposes. Hence, only the count of such reboot is given. Reboots due to other reasons are the main intent. Hence, the reboot time and reboot reason are displayed for other reasons. Reboot time displays the time when the vManage NMS retrieved the reboot data from the device.

API call used to get this details is:

https://vmanage_ip_address $/dataservice/device/reboothistory/synced?deviceId =$

Device-Id is the device's system IP for which reboot history is required

Note— If the device is down for 90 seconds or longer, the reason shows as 'unknown'

3.3.1.16 Crash information

This card gives the information about the number of times the device has crashed. It has two columns, crash time and core file name. Crash time is the time when crash happened and core file name is the name of the core file created as the result of crash. This file name is required to generate the crash log.

API call used to get the details is:

https://vmanage_ip_address $/dataservice/device/crashlog?deviceId =$

DeviceId is the device's system IP for which crash information is required

3.3.1.17 Control Connections

This card displays information about all control connections that the device has with other controller devices in the network. Details are displayed in the table format with columns as, Peer type, peer system IP, peer protocol, site ID, Private IP, Public IP, and local color.

API call used is:

https://vmanage_ip_address $/dataservice/device/control/connections?deviceId =$

DeviceId is the device's system IP for which control connections is required

3.3.1.18 Interface information

This card displays the information about interfaces configured on a device. Details are displayed in a table form. The table only shows interfaces on which IPv4 addresses are configured. The table shows interface name as 'if name', VPN Id, IP address and Admin status.

API call used is:

https://vmanage_ip_address $/dataservice/device/interface/synced?deviceId =$

DeviceId is the device's system IP for which interface information is required.

3.3.2 Integration of Frontend and BackEnd

When user interacts with the GUI tool, a GET or POST request is sent to the BackEnd. BackEnd receives the request and executes the corresponding function. For a GET request, Frontend doesn't send any argument from its side. But if any BackEnd function call needs arguments like system IP of device then POST request is sent along with the device ID. Each function is dedicated to a particular card in GUI tool and calls the respective API to fetch data. The reply of REST API is then filtered and sent back as response to Frontend in JSON format. The response received is structured accordingly to display. The flow below shows the overall process. Angular is used for Frontend and Flask is used for BackEnd.

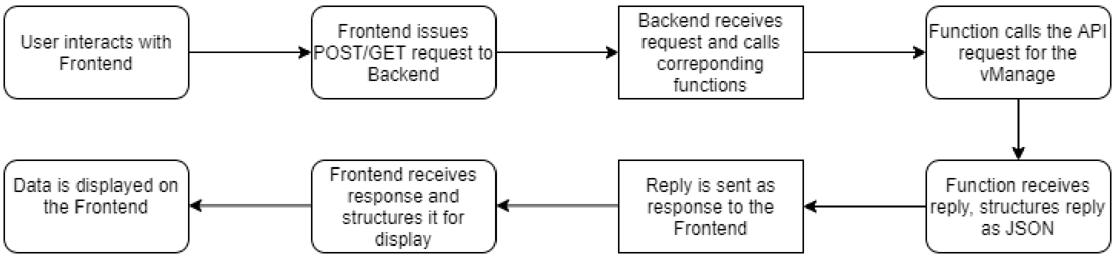

User interacts with Frontend
Frontend issues POST/GET request to Backend
Backend receives request and calls correponding functions
Function calls the API request for the vManage
Function receives reply, structures reply as JSON
Reply is sent as response to the Frontend
Frontend receives response and structures it for display
Data is displayed on the Frontend

FIGURE 3.17: Flowchart for Front End and BackEnd Interaction

3.4 Implementation Practice

Further dependencies were added by installing them via Angular CLI. These packages only enhanced the performance of the application. Performance enhancements included code quality check, syntax error indicators, and efficient functioning. Each of the packages incorporated are describes in detail below:

1. Concurrently – As mentioned earlier, Angular CLI was thoroughly used to create this application. Certain situations may require running several commands simultaneously. Conventionally, you would have top open different terminals on the CLI, for each command. Concurrently avoids this inconvenience and allows multiple commands to be run at once.

2. Prettier – a npm package that facilitates opinionated code formatting. It parses every line of code, and if necessary, makes consistent changes as per its conventions. It also wraps the code when required.

3. Commitizen – This is a command line tool that helps refactor the commit messages by adding lots of pointers. These pointers may be in the form of questions. On commiting a change, these are automatically prompted.

4. Linting – a npm package that parses through the code, to check for potential errors. It is a code quality regulator.

5. Husky – Ideally, all the above mentioned packages must run before a commit is made. Husky is a pre-commit hook that establishes all the tasks to be carried out before a commit. Therefore, once the commit command is administered, husky intervenes to allow functioning of other packages.

Chapter 4

RESULTS, DISCUSSIONS AND CONCLUSIONS

Here, the results of the project work (literature survey and review along with actual work) shall be listed and discussed in detail with appropriate arguments (result analysis) leading to logical conclusions. The list of conclusions should sync with the project objectives. The scope for future research and development in the field of the current project work must also be included in this chapter.

4.1 Results & Analysis

The application, after several revisions, was able to include all criteria set by the experts. The following are the highlights of the dashboard application:

1. At a glance, the dashboard reflects graphical statistics of the most important PKI of a project. This includes Build Rates, Test Cases, Unit Tests, Defects, and Release information.

2. The system also incorporates Business Intelligence, such that the latest data is presented as per access to the dashboard. For instance, if accessed in the begin-ning of the week, the dashboard would reflect all statistics of the previous week.

An access in the mid or latter half of the week would reveal all details until the timestamp of access.

3. The system is well adapted and tested to handle real time data, with instant changes in graphical projections. The system is trained to update these changes every mid-day and accordingly reflect them on the dashboard.

4. Efficient space management of the limited tile size is facilitated through the carousel feature that encapsulates all the graphs. The carousel component occupies half of the tile, thereby affording the ease to read the charts comfortably.

5. The dashboard is developed to reflect data in the most simplistic and perceivable format. Continuous revisions in design allowed for a very insightful representation of the projects' PKIs.

4.2 Comparative Study

There are a lot of tools as well as platforms available today to create similar applications.

A few to name are Grafana and Kibana dashboards. The same were consulted in order to get an innate understanding of the topic in hand. Each of the DevOps Dashboards had a multitude of options, that were closely reviewed, so that only the best was incorporated into the application. After the completion of the development process, here are a few aspects that stand out from other dashboards:

1. The application provides for an option to alter the graphical projections and data as per a prescribed period. The options include time periods in days, weeks and months. Enabling any of the options allows for a customized transformation of the data.

2. As mentioned earlier, the project tile accommodates a carousel that loops through three graphs. This is an absolute unique feature built into the application.

3. The dashboard allows the user two glimpses into a project. A short and crisp summary as well as a detailed and exaggerated report. Both of these options are easily accessible and stand as distinct features when compared to other similar applications.

4.3 Discussions

The dashboard is developed to accommodate and reflect all statistics of projects available in the company's internal DVCS. This application is strictly to stay within the limits of the organization and be used alongside their DVCS, for a more guided development process.

4.4 Conclusions

This project was created, keeping in mind, the requirements and quality standards of the company. The creation of a Dashboard is challenging and commands focused efforts. The application has immense potential and an ability to issue well-planned business ideas, that will reshape the SDWAN. Despite the involved complexity, the application development and all its aspects were thoroughly enjoyed. The number of things learnt in the period is simply not quantifiable. The development of this application has triggered an ardent interest in the domain ‘Web Development’ and opened a window of opportunity to pursue it. Most importantly, an exposure to the quality standards of the industry and its work culture was earned.

4.5 Scope for Future Work

The application is scalable and adaptable to include many features that could enhance its performance. The dashboard has potential to fully transform as per every individual user. This would promote better outreach and understanding of its contents. Similar user-specific customizations, priority setting and alerts could escalate its functionality and importance. Also, the current design caters to a fully furnished frontend connected to a fake REST API. Efforts can now be dedicated to administer the backend and trans form it to an accomplished MEAN stack application.

Bibliography

[1] K. Yang, D. Guo, B. Zhang and B. Zhao, *Multi-Controller Placement for Load Balancing in SDWAN*, IEEE Access, vol. 7, pp. 167278-167289, 2019.

[2] K. Yang, D. Guo, B. Zhang and B. Zhao, *Load Deployment Algorithm in Hierarchical Architecture for SDWAN*, IEEE Access, vol. 7, pp. 65839-65851, 2019.

[3] K. Yang, B. Zhang, D. Guo, M. Lin and T. de Cola, *Partitioned Controller Placement in SDWANs for Reliability Maximization with Latency Constraints*, 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 2019, pp. 1-6.

[4] V. Lanka and H. K. Mohan, *NETWORK IN A BOX - Automation of secure SD-WAN with Service chaining in Juniper NFX*, 2019 11th International Conference on Communication Systems and Networks (COMSNETS), Bengaluru, India, 2019, pp. 527-529.

[5] R. Liu, S. Li, H. Wang and Z. Tang, *QoS Routing Optimization Algorithm Based on Hierarchical Multi-Controller Coordination*, 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chengdu, China, 2019, pp. 1820-1825.

[6] Z. Yang, Y. Cui, B. Li, Y. Liu and Y. Xu, *Software-Defined Wide Area Network (SD-WAN): Architecture, Advances and Opportunities*, 2019 28th International Conference on Computer Communication and Networks (ICCCN), Valencia, Spain, 2019, pp. 1-9.

[7] A. C. Houle, L. Boulianne and L. Dupras, *SD-WAN : A Technology for the Efficient Use of Bandwidth in Multi-Wavelength Networks*, OFC/NFOEC 2008 - 2008 Conference on Optical Fiber Communication/National Fiber Optic Engineers Conference, San Diego, CA, 2008, pp. 1-10.

[8] R. E. Mora-Huiracocha, P. L. Gallegos-Segovia, P. E. Vintimilla-Tapia, J. F. Bravo-Torres, E. J. Cedillo-Elias and V. M. Larios-Rosillo, *Implementation of a SD-WAN for the interconnection of two software defined data centers*, 2019 IEEE Colombian Conference on Communications and Computing (COLCOM), Barranquilla, Colombia, 2019, pp. 1-6.

[9] L. Vdovin, P. Likin and A. Vilchinskii, *Network utilization optimizer for SD-WAN*, 2014 International Science and Technology Conference (Modern Networking Technologies) (MoNeTeC), Moscow, 2014, pp. 1-4.

[10] R. Subramanian and J. Puthenparambil, *Shared Memory Enabled Service Plane Optimization*, 2020 International Conference on COMmunication Systems NETworkS (COMSNETS), Bengaluru, India, 2020, pp. 683-684.

[11] C. N. Sminesh, E. G. M. Kanaga and A. Roy, *Optimal multi-controller placement strategy in SD-WAN using modified density peak clustering*, IET Communications, vol. 13, no. 20, pp. 3509-3518, 19 12 2019.

[12] Z. Duliński, R. Stankiewicz, G. Rzym and P. Wydrych, *Dynamic Traffic Management for SD-WAN Inter-Cloud Communication*, IEEE Journal on Selected Areas in Communications

[13] V. Lanka and H. K. Mohan, *NETWORK IN A BOX - Automation of secure SD-WAN with Service chaining in Juniper NFX*, 2019 11th International Conference on Communication Systems and Networks (COMSNETS), Bengaluru, India, 2019, pp. 527-529.

[14] R. Liu, S. Li, H. Wang and Z. Tang, *QoS Routing Optimization Algorithm Based on Hierarchical Multi-Controller Coordination*, 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chengdu, China, 2019, pp. 1820-1825.

Appendix A

Mean Stack

MEAN stack is a full stack JavaScript framework that is utilised for the development of dynamic and responsive applications. It provides developers with an organised and simple procedure to build rapid prototypes. Its unique feature lies in the fact that it employs only one language at every level of application - JavaScript. This makes it efficient and gives it a modern take on web development. Some of the advantages of MEAN stack development are:

- It is simple and straightforward
- The stack is adaptable across a wide range of applications.
- Builds applications with multi-user interactivity.
- Helps in the creation of real-time systems.

A.1 Components

It is a free and open source software bundle that encapsulates 4 components:

1. MongoDB - No SQL database that stores and manages the application's data. It is recognised for its scalability in terms of storage and performance.

2. ExpressJS - an application framework for NodeJS, it handles all interactions between the frontend and DB and facilitates seamless data transfer.

3. Angular - the popular JavaScript front-end framework is used to develop the UI of the application.

4. NodeJS - server-side JavaScript framework, forms the backbone of the stack by processing multiple requests asynchronously.

A.2 Working

When the user makes a request, client-side processing is undertaken by Angular. NodeJS processes the client-server requests and pushes it to ExpressJS. ExpressJS communicates with the DB which in turn responds with the retrieved data. This data is returned to NodeJS and then made available at the front-end.